SCENARIserver4.1 : manuel d'installation et d'administration technique (Linux)

SCENARIserver4.1 : manuel d'installation et d'administration technique

Juillet 2015 Version : 1.2 Kelis

SCENARIserver4.1 : manuel d'installation et d'administration technique

Version : 1.2

Date : Juillet 2015

Editeur : Kelis

Auteur(s) : Antoine Pourchez, Laurent Riviere, Samuel Monsarrat

Copyright : Kelis

Licence :

Cette œuvre est mise à disposition selon les termes de la Licence Creative Commons Attribution 3.0 France [http://creativecommons.org/licenses/by/3.0/fr/] .

Table des matières

Introduction 4
Chapitre 1 - Architecture technique 6
Chapitre 2 - Installation des outils tiers 7
1. Installation d'une JVM 7
2. Installation d'un moteur de servlet 7
3. Installation de LibreOffice 4.3 ou plus (Optionnelle) 8
4. Installation de Latex (Optionnelle) 9
5. Compléments 9

Chapitre 3 - Configuration système 10
Chapitre 4 - Configuration de Tomcat 13
Chapitre 5 - Installation/Configuration de SCENARIserver 22
1. Configuration de SCENARIserver 22
2. Installation 26
3. Montée de version 29

Chapitre 6 - Installation/Configuration de SCENARIserver sous Debian 32
1. Installation 32
2. Configuration 35

Chapitre 7 - Monitoring 41
Chapitre 8 - SCENARIbatch 43
1. Installation 43
2. Sauvegarde / Restauration [DB] 43

Chapitre 9 - Proxy [DB] 47

Introduction

Conventions

Dans ce document, nous utiliserons les conventions suivantes :

- [xxx.xxx.xxx.xxx] : adresse IP du serveur, ou URL d'accès ;
- [pathData] : path absolu d'accès au répertoire de travail de l'application.

SCENARIserver vs SCENARIserverLite

Depuis la version 4.0 de SCENARIserver, deux branches sont proposées :

- **SCENARIserverLite** : les fichiers de contenu utilisateur sont enregistrés sur le disque dur du serveur sous la même forme que leur vue dans l'atelier. L'administration est simplifiée, mais les fonctionnalités limitées.
- **SCENARIserver** : cette application intègre un système de base de données orienté graphe de documents. Les fichiers de contenu utilisateur sont enregistrés dans un ensemble de fichiers binaires (blobs).

  IMPORTANT : Cette application, au spectre fonctionnel complet (gestion des fonctions collaboratives, ...), impose une administration matérielle et logicielle rigoureuses (stabilité du système, gestion des backups, monitoring, ...) au risque de compromettre l'intégrité des données.

Spécificités SCENARIserver

> Les parties spécifiques à SCENARIserver (non applicables à SCENARIserverLite) sont indiquées par la marque **[DB]**, DB pour database.
>
> Les parties spécifiques à SCENARIserverLite (non applicables à SCENARIserver) sont indiquées par la marque **[Lite]**.

Ce manuel d'installation et d'administration technique s'adresse à un profil **Administrateur**, en charge de l'installation, surveillance et administration technique de SCENARIserver.

Pré-requis

> Avant toute manipulation sur votre SCENARIserver, assurez-vous d'avoir effectué une **sauvegarde** et vérifié son intégrité.
>
> Kelis ne saurait être tenu pour responsable de tout dommage direct ou indirect résultant de l'application de ces procédures, conformément à la licence du présent document.

Debian / Ubuntu

> SCENARIserver est disponible dans les dépôts Debian de scenari-platform.org. Si vous voulez installer SCENARIserver de cette manière, reportez-vous à `Installation/Configuration de SCENARIserver sous Debian` *(cf. Installation/Configuration de SCENARIserver sous Debian)* [p. 32]

Documentation papier

La version papier de cette documentation est disponible ici [http://docs.kelis.fr/sc41/adminTech/lin/paper/guidePapier_lin.pdf] .

Chapitre 1

Architecture technique

L'environnement applicatif visé est de type client/serveur 2-tiers. Il est composé de :

- **SCENARIserver** : webapp basée sur Java à installer dans un serveur d'application (Tomcat) ;
- **SCENARIclient** : client riche de type RDA (Rich Desktop Application) qui s'installe comme une application standalone.

La communication entre ces deux niveaux applicatifs se fait via le protocole http, ou https.

Spécifications matérielles

Les spécifications matérielles requises pour SCENARIserver dépendent beaucoup du nombre d'utilisateurs, du volume de contenu à produire, de l'usage simultané du système de publication, et du modèle documentaire.

Les chiffres suivants sont fournis à titre indicatif. Pour le modèle Opale, un nombre d'utilisateurs de 20 qui ont un usage normal de la publication (pas de publication toutes les 5 minutes), on pourrait recommander les spécifications suivantes :

- **Processeur** quad-core >= 2,5Ghz ;
- **RAM** : 4Go ou plus recommandé ;
- **DD** : 1Go d'espace disque disponible pour effectuer l'installation (50Go d'espace disque disponible recommandé, la taille du disque dur dépend beaucoup du type de contenu qui sera à traiter : les ressources multimédia occupent beaucoup plus d'espace disque que les ressources textuelles).
- **OS Windows** (Expériences avec : Windows Server 2008 R2, Windows Server 2012)

  OS Linux (Expériences avec Debian 7+ (Tomcat 7 et 8), Ubuntu 14.04+ (Tomcat 7 et 8), Redhat/Centos 6+ (Jboss, Tomcat 7 et 8)

Téléchargement

- Windows : http://scenari-platform.org/projects/scenari/files/SCENARIserver/win/latestStable/
- Linux 32bits : http://scenari-platform.org/projects/scenari/files/SCENARIserver/lin/latestStable/
- Linux 64bits : http://scenari-platform.org/projects/scenari/files/SCENARIserver/lin_x86_64/latestStable/

Chapitre 2

Installation des outils tiers

Vous trouverez des exemples de script d'installation sur cette page : http://scenari-platform.org/trac/scenari/wiki/InstallServerLinux

1. Installation d'une JVM

SCENARIserver doit tourner dans une JVM d'Oracle JDK/JRE 7 ou OpenJDK version 7.
Vous pouvez utiliser la JVM proposée par votre distribution, ou bien en télécharger une.

gij

> SCENARIserver EST INCOMPATIBLE avec gij (GNU Interpreter for Java).

Installation de la JVM d'Oracle (Java SE 7)

Suivre la documentation d'installation fournie par Oracle pour Java SE 7.

Source de l'installeur : http://www.oracle.com/technetwork/java/javase/archive-139210.html [http://www.oracle.com/technetwork/java/javase/archive-139210.html]

Sous Debian et Ubuntu

```
sudo add-apt-repository -y ppa:webupd8team/java
sudo apt-get update && sudo apt-get install oracle-java7-set-default
```

2. Installation d'un moteur de servlet

Scenari est packagé de façon standard (war), permettant un déploiement sur divers serveurs d'application.
Néanmoins, les installations sur Tomcat sont les plus courantes aujourd'hui, et sont donc conseillées.

Tomcat (conseillé)

Installation de Tomcat

Suivre la documentation d'installation fournie par Apache.

Sources de l'installeur Tomcat 7 : http://tomcat.apache.org/download-70.cgi.
Sources de l'installeur Tomcat 8 : http://tomcat.apache.org/download-80.cgi.

Installation

L'installation de Tomcat passe par la décompression du fichier téléchargé sur le disque dur (dans le répertoire `[pathApp]/Tomcat7` par exemple).

> Le path d'installation de Tomcat ne doit contenir que des caractères alphanumériques (bug présent dans la JVM 1.6).
>
> ***Si vous ne respectez pas cette recommandation, svous risquez de tomber sur cette erreur 'java.net.URISyntaxException: Illegal character in path ...'.***

> Le path d'installation sera appelé "`[TOMCAT_HOME]`" dans le suite du document.

JBoss

Installation de JBoss

Suivre la documentation d'installation fournie par JBoss.
Sources de l'installeur JBoss : http://jbossas.jboss.org/downloads/

> Le path d'installation sera appelé "`[JBOSS_HOME]`" dans le suite du document.

3. Installation de LibreOffice 4.3 ou plus (Optionnelle)

Suivre la documentation d'installation fournie par The Document Foundation.
Source de l'installeur : https://www.libreoffice.org/download/libreoffice-still/

Les polices

> Il faut avoir sur votre système les Microsoft Core Fonts (voir les Compléments).

Ubuntu Server et LibreOffice 4.3

> Depuis LibreOffice 4.3 sous un Ubuntu Server (12.04 ou plus), il faut avoir les packages "libgl1-mesa-glx" et "libglu1-mesa".

4. Installation de Latex (Optionnelle)

L'exploitation des formules mathématiques LaTeX impose la présence de Tex Live sur le serveur :

Source de l'installeur : http://www.tug.org/texlive/

Vous trouverez des informations complémentaires sur cette page : http://scenari-platform.org/trac/modelet/wiki/math/latex

Sous Ubuntu et Debian

sudo apt-get install texlive-latex-base texlive-latex-recommended texlive-fonts-recommended texlive-latex-extra preview-latex-style dvipng

Sous Redhat et Centos

yum install texlive-latex texlive-dviutils

5. Compléments

Polices

Pour certains modèles de document ou pour intégrer des ressources LibreOffice, il est vivement recommandé d'installer le package "msttcorefonts"" de votre distribution Linux.

Sous Debian et Ubuntu

sudo apt-get install ttf-mscorefonts-installer

Sous Redhat et Centos

yum install curl cabextract xorg-x11-font-utils fontconfig

rpm -i https://downloads.sourceforge.net/project/mscorefonts2/rpms/msttcore-fonts-installer-2.6-1.noarch.rpm

Chapitre 3

Configuration système

Variables d'environnement

Les variables suivantes doivent être déclarées (adapter les path) :

- **CATALINA_HOME** ***(si le serveur d'application Tomcat a été choisi)*** : chemin absolu vers apache Tomcat (ex : `/opt/apache-tomcat-7.0.64`) ;
- **JAVA_HOME** : chemin absolu vers la JDK précédemment installée (ex : `/opt/jdk1.7.0_80`) ;
- **OOO_DISABLE_RECOVERY**=1 ***(si libreOffice a été installé)*** : configuration du mode de reprise sur incident de LibreOffice ;
- **UNO_PATH** ***(si libreOffice a été installé)*** : chemin vers le répertoire program de LibreOffice (ex : `/opt/libreoffice4.3/program`).

Mémoire swap

L'application SCENARIserver est amenée à exécuter des applications tierces (LibreOffice, convert etc).

L'implémentation sous Unix (Linux, bsd etc) de JAVA à un limitation sérieuse qui dicte que lorsqu'un processus Java démarre un nouveau processus (effectue un fork), le système doit avoir au moins la même quantité de mémoire disponible que la quantité occupée par le processus Java.

Si l'OS ne dispose pas d'assez de mémoire l'exception suivante est levée :

```
java.io.IOException: java.io.IOException: Cannot allocate memory
at java.lang.UNIXProcess.<init>(UNIXProcess.java:148)
at java.lang.ProcessImpl.start(ProcessImpl.java:65)
at java.lang.ProcessBuilder.start(ProcessBuilder.java:451)
at java.lang.Runtime.exec(Runtime.java:591)
```

cf :

- http://developers.sun.com/solaris/articles/subprocess/subprocess.html ;
- http://stackoverflow.com/questions/209875/from-what-linux-kernel-libc-version-is-java-runtime-exec-safe-with-regards-to-m ;
- http://www.coderanch.com/t/419196/Java-General/java/there-any-way-execute-Linux ;

> Si le processus Java de SCENARIserver utilise 900MB de mémoire, au moment ou SCENARI démarre l'application convert pour redimensionner un image, il faut que le système dispose d'au moins 900MB de mémoire allouable (RAM et swap confondu). Les 900MB supplémentaires ne seront jamais réellement alloués (convert utilise en moyenne 3MB) mais pendant un cours instant il faut qu'ils soient présents.

Contournement du problème : ajouter de la mémoire swap

Une solution à ce problème est simplement d'ajouter de la mémoire swap supplémentaire, en générale d'au mois la quantité de mémoire alloué au processus Java.

La procédure suivante ajoute un fichier de swap supplémentaire de 1Gb :

- Dans notre exemple on ajoute un ficher vide dans "/" :

```
$ sudo dd if=/dev/zero of=/swap.img bs=1024k count=1024
```

- Le formater en swap :

```
$ sudo mkswap /swap.img
```

- Ajouter le swap au système en cours d'exécution :

```
$ sudo swapon /swap.img
```

- Vous pouvez ajouter ce fichier à fstab pour qu'il soit ajouté automatiquement au démarrage :

```
$ sudo nano /etc/fstab
```

- Ajouter la ligne suivante à la fin du fichier

```
/swap.img none swap sw 0 0
```

Limite du nombre de fichiers ouverts simultanément

L'application SCENARIserver est amené à ouvrir un grand nombre de fichier simultanément.

Par défaut Linux impose une limite au nombre de fichiers ouvrable par un même utilisateur, en général cette limite est de 1024 fichiers.

Il convient donc d'augmenter cette limite pour l'utilisateur qui fait tourner la JVM de Tomcat sinon vous risquez de tomber sur l'erreur Java "Too many open files".

Augmentation de cette limite à 4096 sur Debian/Ubuntu

- Ouvrir le fichier `/etc/security/limits.conf` et ajoutez les lignes :

```
<user> soft nofile 4096
<user> hard nofile 4096
```

ou <user> est l'utilisateur qui doit faire tourner SCENARIserver.

- Indique au Kernel qu'il doit recharger sa configuration :

```
#sysctl -p
```

- Activer la prise en compte des limites de /etc/security/limits.conf par PAM

  Ouvrir le fichier /etc/pam.d/su et dé-commenter la ligne :

```
session required pam_limits.so
```

Au prochain démarrage de SCENARIserver, le nombre maximal de fichiers ouverts simultanément sera de 4096.

- http://ubuntuforums.org/showthread.php?t=902873;
- http://lj4newbies.blogspot.com/2007/04/too-many-open-files.html;

Chapitre 4

Configuration de Tomcat

Pour permettre à des webapps SCENARIserver de fonctionner correctement dans le serveur de servlets Tomcat il est nécessaire d'y apporter certaines modifications de paramétrage.

Sécurité

A l'heure actuel, Les webapps SCENARIserver ne sont pas compatible avec le "Security Manager" de Tomcat. Il est important de ne pas activer le module "Security Manager" lors du démarrage de Tomcat.

Par défaut le "Security Manager" n'est pas actif, cependant il faut être vigilant sur ce point lors de l'usage de versions packagées de Tomcat sous Linux.

Usage d'un proxy Apache

Il est possible de cacher Tomcat derrière un proxy Apache. Ceci permet entre autres d'accéder à SCENARIserver sur le port 80 même si vous avez déjà une serveur web Apache d'installé. Il y a plusieurs façons de mettre en place un proxy Apache vers Tomcat, le plus simple et le seul qui à été testé avec succès avec SCENARIserver est l'usage du module Apache mod_proxy :

- HOWTO de la documentation Tomcat 7 [http://tomcat.apache.org/tomcat-7.0-doc/proxy-howto.html] ou HOWTO de la documentation Tomcat 8 [http://tomcat.apache.org/tomcat-8.0-doc/proxy-howto.html] ;
- Documentation du module Apache mod_proxy [http://httpd.apache.org/docs/2.2/mod/mod_proxy.html].

Tomcat sous Debian / Ubuntu

Si vous utilisez la version packagée de Tomcat, tous les fichiers de configuration de Tomcat sont dans `/etc/tomcat6`.

Il convient alors de remplacer les chemins `[TOMCAT_HOME]/conf` dans les procédures ci-dessous par `/etc/tomcat6/conf`.

Paramétrer la mémoire RAM allouée à l'application

- Si Tomcat est lancé, le stopper (cf procédure dans le présent guide) ;
- Ouvrir le fichier `[TOMCAT_HOME]/bin/catalina.sh` ;
- Créer la variable JAVA_OPTS en début de fichier ;

Exemple pour 1024Mo

```
JAVA_OPTS="-Xms1024M -Xmx1024M $JAVA_OPTS"
```

- Procédure alternative

Paramétrer la mémoire RAM allouée à Tomcat dans Debian

- Ouvrir le fichier `/etc/defaults/tomcat6`
- Modifier la variable JAVA_OPTS en milieu de fichier ;

  Adapter l'option ***-Xmx*** et ajouter l'option ***-Xms***

  Exemple pour 1024Mo

  ```
  JAVA_OPTS="-Xms1024M -Xmx1024M $JAVA_OPTS"
  ```

- Relancer le serveur d'application.

  Pour Tomcat 6 :

  ```
  user@computer:~$ sudo service tomcat6 restart
  ```

Charger les librairies LibreOffice

L'utilisation de plusieurs webApp SCENARI sur le même serveur Tomcat nécessite un paramétrage permettant le partage de l'accès à LibreOffice.

- Ouvrir le fichier `[TOMCAT_HOME]/conf/catalina.properties` ;
- Modifier la ligne "shared.loader" pour y déclarer les chemins des librairies LibreOffice `jurt.jar`, `ridl.jar` et `unoil.jar`.

  Exemple de configuration windows

  ```
  shared.loader=file:///D:/Program Files (x86)/LibreOffice 4/URE/java/jurt.jar,file:///D:/Program Files (x86)/LibreOffice 4/URE/java/ridl.jar,file:///D:/Program Files (x86)/LibreOffice 4/Basis/program/classes/unoil.jar
  ```

  Exemple de configuration linux

  ```
  shared.loader=/opt/libreoffice4.3/ure/share/java/jurt.jar, /opt/libreoffice4.3/ure/share/java/ridl.jar,/opt/libreoffice4.3/program/classes/unoil.jar
  ```

Connecteurs Tomcat

Le connecteur Tomcat doit être utilisé pour les spécification suivantes :

- Spécification du port (8080 par défaut) ;
- Encodage des URLs ;
- Taille des buffers ;
- ...

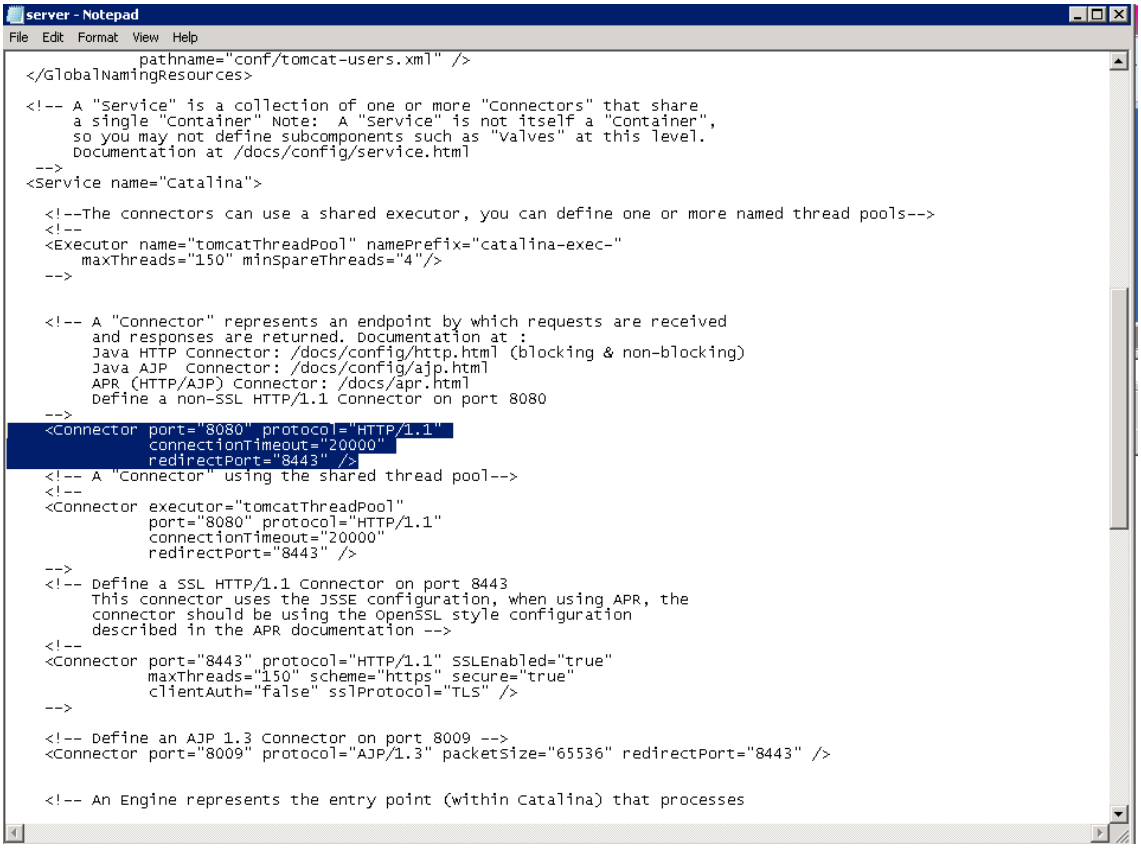
Pour éviter des problèmes d'encodage de nom de fichier il est important de forcer Tomcat à traiter toutes les url en UTF-8.

- Ouvrir le fichier `[TOMCAT_HOME]/conf/server.xml` avec un éditeur XML (type notepad);

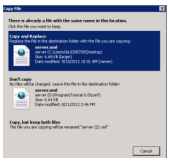
> Assurez-vous de bien disposer des droits d'écriture sur ce fichier. Si tel n'est pas le cas, et que vous disposez des droits de remplacement uniquement, copier/coller au préalable le fichier sur votre bureau le temps de le modifier.

- Modifier ou ajouter un *Connector* sur le protocole HTTP/1.1.
  - Pour modifier le port, utiliser l'attribut `port` ;
  - Pour éviter des problèmes d'encodage de nom de fichier il est important de forcer Tomcat à traiter toutes les url en UTF-8. Ajouter donc le paramètre `URIEncoding="UTF-8"`
  - Pour étendre la taille autorisée des entêtes HTTP, ajouter le paramètre `maxHttpHeaderSize="65536"`

  Exemple complet de paramétrage : `<Connector port="8080" protocol="HTTP/1.1" connectionTimeout="20000" redirectPort="8443" URIEncoding="UTF-8" maxHttpHeaderSize="65536"/>`



> [facultatif] Si le fichier `server.xml` a été copié au préalable sur le bureau pour y effectuer les modifications, penser à le remettre dans le répertoire `[TOMCAT_HOME]/conf/server.xml`.



Suppression des éléments non utilisés de Tomcat

Tomcat apporte un lot d'outils lors de son installation standard non utilisés, ou qui pourraient apporter des faille de sécurité. Il est donc conseillé de les supprimer.

- Supprimer les répertoires "docs", "examples", "host-manager", "manager " dans [TOMCAT_HOME]/webapps.

  Purge des webapps par défaut de Tomcat

- Ouvrir le fichier [TOMCAT_HOME]/conf/server.xml, et supprimer le connector "AJP "

  Supprimer ou commenter la ligne `<Connector port="8009" protocol="AJP/1.3" redirectPort="8443" />`

Démarrer Tomcat

Cette procédure décrit comment démarrer le serveur d'application Tomcat.

Conditions d'utilisation

Tomcat a été paramétré pour démarrer automatiquement lors de l'initialisation du serveur. Cette procédure n'est donc à utiliser qu'en cas de dysfonctionnement rencontré sur le serveur d'application.

- Exécuter la commande `[TOMCAT_HOME]/bin/catalina.sh start`

  Il faut être connecté avec votre utilisateur pour tomcat pour réaliser cette action

  ### Test de bon fonctionnement

  Une fois démarré, la requête suivante doit répondre http://127.0.0.1:8081/scserver40/s/chain/u/ping (page blanche).

Stopper Tomcat

Cette procédure décrit comment arrêter le serveur d'application Tomcat.

- Exécuter la commande `[TOMCAT_HOME]/bin/catalina.sh stop` ;

  Il faut être connecté avec votre utilisateur pour tomcat pour réaliser cette action

Passer Tomcat (SCENARIserver) en service

- Copier-coller le code suivant dans un éditeur de texte

  ### Red Hat, Fedora, CentOS

```
1 # This is the init script for starting up SCENARIserver
2 #
3 # chkconfig: 345 90 10
4 # description: Starts and stops the Tomcat daemon of SCENARIserver.
5 #
6
7
  #============================================================================
8 # Server configuration
```

```
9
10 # Name of the server (used when printing out status).
11 serverName=SCENARIserver
12
13 # Set this to the path of the tomcat folder.
14 tomcat=/opt/SCENARIserver/tomcat
15
16 # user to run the tomcat server as
17 # (you have to create this user yourself, he must be owner or have write permission
18 # on tomcat webapp dir, scenari workshop dir and other temporary dirs)
19 runasuser=root
20
21 # Set this if you need to specify a specific JRE.
22 export JAVA_HOME=/opt/SCENARIserver/jre
23
24 # SCENARIserver needs a minimum of 256Mb of heap space.
25 heapspace=512M
26
27 # Extra JVM options
28 extra_options=""
29
30 # JVM options
31 opts_start=" -Xms$heapspace -Xmx$heapspace -Dorg.apache.tomcat.util.buf.UDecoder.ALLOW_ENCODED_SLASH=true $extra_options"
32 opts_stop=" -Xms64M -Xmx128M "
33
34 # Path to tomcat pid file.
35 export CATALINA_PID="$tomcat/tomcat.pid"
36
37 # Shutdown timeout before tomcat is killed (in seconds)
38 kill_timeout=30
39
40
#================================================================================
41 # LibreOffice Configuration
42
43 # Set DISPLAY variable if you want LibreOffice to use a specific display
44 export DISPLAY="localhost:1.0"
45
46 # If set this stops the recovery dialog prompting you as OO.o starts up
47 # after a crash - instead the recovery files are just silently
48 # accumulated.
49 export OOO_DISABLE_RECOVERY=1
50
51
#================================================================================
52
53 # Source function library.
54 . /etc/rc.d/init.d/functions
55
56 # Get config.
57 . /etc/sysconfig/network
58
59 # Check that networking is up.
60 [ "${NETWORKING}" = "no" ] && exit 0
61
62 startup=$tomcat/bin/startup.sh
63 shutdown=$tomcat/bin/shutdown.sh
64
65 start() {
```

```
    [ -x $startup ] || exit 5
    # Tomcat pid checks
    if [ -f $CATALINA_PID ] ; then
     if [ "$(ps -p `cat $CATALINA_PID` | wc -l)" -gt 1 ]; then
        # Tomcat process is still running
        echo "ERROR: $serverName seems to be already running as process `cat $CATALINA_PID`"
        exit 1
      else
        # Tomcat process not running, but lock file not deleted?
        echo "WARNING: orphan lock file. Lock file deleted."
        rm $CATALINA_PID
      fi
    fi
    echo -n $"Starting $serverName: "
    export JAVA_OPTS="$opts_start"
    su -s /bin/bash -c $startup $runasuser
    retval=$?
    echo
    return $retval
}

stop() {
    echo -n $"Stopping $serverName: "
    # Kill all LibreOffice instances run but the tomcat user...
    pkill -9 -u $runasuser soffice
    export JAVA_OPTS="$opts_stop"
    su -s /bin/bash -c "$shutdown $kill_timeout -force" $runasuser
    retval=$?
    echo
    return $retval
}

restart(){
  stop
  start
}

# See how we were called.
case "$1" in
start)
 start
 ;;
stop)
 stop
 ;;
status)
# This doesn't work ;)
# status SCENARIserver
 ;;
restart)
 restart
 ;;
*)
 echo $"Usage: $0 {start|stop|restart}"
 exit 1
esac

exit 0
```

Debian, Ubuntu, Mint

```
#! /bin/bash
# This is the init script for starting up SCENARIserver.
#
# description: Starts and stops the Tomcat daemon of SCENARIserver
#


#=============================================================================
# Server configuration

# Name of the server (used when printing out status).
serverName=SCENARIserver

# Set this to the path of the tomcat folder.
tomcat=/opt/SCENARIserver/tomcat

# user to run the tomcat server as
# (you have to create this user yourself, he must be owner or have write permission
# on tomcat webapp dir, scenari workshop dir and other temporary dirs)
runasuser=root

# Set this if you need to specify a specific JRE.
export JAVA_HOME=/opt/SCENARIserver/jre
export PATH="$PATH:/usr/bin"

# SCENARIserver needs a minimum of 256Mb of heap space.
heapspace=512M

# Extra JVM options
extra_options=""

# JVM options
opts_start=" -Xms$heapspace -Xmx$heapspace -Dorg.apache.tomcat.util.buf.UDecoder.ALLOW_ENCODED_SLASH=true $extra_options"
opts_stop=" -Xms64M -Xmx128M "

# Path to tomcat pid file.
export CATALINA_PID="$tomcat/tomcat.pid"

# Shutdown timeout before tomcat is killed (in seconds)
kill_timeout=30


#=============================================================================
# LibreOffice Configuration

# Set DISPLAY variable if you want LibreOffice to use a specific display
export DISPLAY=":1.0"

# If set this stops the recovery dialog prompting you as OO.o starts up
# after a crash - instead the recovery files are just silently
# accumulated.
export OOO_DISABLE_RECOVERY=1

export UNO_PATH=/usr/lib/libreoffice/program/


#=============================================================================

```

```
startup=$tomcat/bin/startup.sh
shutdown=$tomcat/bin/shutdown.sh

. /lib/lsb/init-functions

start(){
    # Tomcat pid checks
    if [ -f $CATALINA_PID ] ; then
     if [ "$(ps -p `cat $CATALINA_PID` | wc -l)" -gt 1 ]; then
        # Tomcat process is still running
        echo "ERROR: $serverName seems to be already running as process `cat $CATALINA_PID`"
        exit 1
      else
        # Tomcat process not running, but lock file not deleted?
        echo "WARNING: orphan lock file. Lock file deleted."
        rm $CATALINA_PID
      fi
    fi
    log_begin_msg "Starting $serverName..."
    export JAVA_OPTS="$opts_start"
    su -s /bin/bash -c $startup $runasuser
    log_end_msg $?
}
stop(){
    log_begin_msg "Stoping $serverName..."
    export JAVA_OPTS="$opts_stop"
    # Kill all LibreOffice instances run but the tomcat user...
    pkill -9 -u $runasuser soffice
    su -s /bin/bash -c "$shutdown $kill_timeout -force" $runasuser
    log_end_msg $?
}
restart(){
    stop
    start
}

# See how we were called.
case "$1" in
start)
    start
    ;;
stop)
    stop
    ;;
status)
    # status SCENARIserver
    echo "Status: not implemented"
    ;;
restart)
    restart
    ;;
*)
    echo "Usage: $0 {start|stop|restart}"
    exit 1
esac

exit 0
```

- ÉditeR les lignes indiquant :

- le répertoire de tomcat, (variable tomcat) ;
- le répertoire de la JRE, (variable JAVA_HOME) ;
- le répertoire de libreoffice, (variable UNO_PATH) ;
- les options de lancement de la JRE : la mémoire alloué à tomcat dans cet exemple est 512M. Adaptez les valeurs -Xms et -Xmx à votre contexte ;
- le display utilisable pour libreoffice ;

Installation

- Sauver le fichier dans /etc/init.d sous le nom SCENARIserver (ce répertoire est standard sur les distributions récentes)
- Rendre le script exécutable: chmod a+x SCENARIserver
- Ajouter SCENARIserver au bon runlevel :
  - sous Fedora : `chkconfig --add SCENARIserver`
  - sous Ubuntu : `update-rc.d SCENARIserver defaults 90 10`

Chapitre 5

Installation/Configuration de SCENARIserver

SCENARIserver est fourni sous la forme d'une archive compressée.

Une fois décompressé et paramétré, SCENARIserver sera à compiler pour produire la webApp (fichier .war) à installer dans Tomcat.

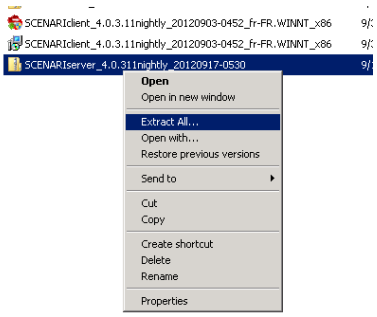
1. Configuration de SCENARIserver

La webApp SCENARIserver est fortement paramétrable en fonction du contexte d'utilisation et des fonctionnalités désirées.

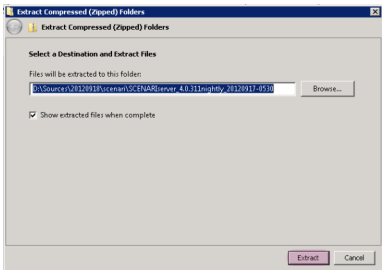
Configurer SCENARIserver

- Décompressez le fichier SCENARIserver_xxx.zip fourni dans le répertoire `[pathSrc]/Vx.y.z/SCENARIserver`.

  Faire un click droit sur le fichier SCENARIserver_xxx.zip.

  Cliquer sur `Extract`.

- Paramétrez les fichiers de configuration du répertoire `[pathSrc]/Vx.y.z/SCENARIserver/conf`.

Fichier "main.property "

[requis] server.webapp.name : Nom de l'application

Cette variable définit le nom de la webApp qui sera déployée. Seuls les caractères alphanumériques sont autorisés.

ex : **server.webapp.name**=scserver41

Chemins d'accès

[requis] server.work.path : Répertoire principal des données

Cette variable définit le répertoire [pathData] utilisé dans le présent document à diverses reprises.

Exemple : **server.work.path**=E:/data/scserver41

[optionnel] server.temp.path : Répertoire de travail (backup inutile)

Cette variable définit le répertoire [pathData] utilisé dans le présent document à diverses reprises.

Exemple : **server.temp.path**=E:/tmp/scserver41

[optionnel] server.gen.path : Répertoire racine de sortie des générateurs

Cette variable définit le répertoire [pathData] utilisé dans le présent document à diverses reprises.

Exemple : **server.gen.path**=E:/data/scserver41/javaserver/gen

[optionnel] server.log.path : Répertoire des logs scenari

Cette variable définit le fichier de log scenari.

Exemple : **server.log.path**=[pathLog]/[webAppName]/scenari_[date].log

Remarque : penser à retirer le caractère # de mise en commentaire de la variable server.log.path

Sécurité

[optionnel] server.conf.enhancedSecurity : Activation de la sécurité avancée

Cette variable permet d'activer un niveau de sécurité accru

Remarque : consommateur de ressources

Configuration du mode batch

[optionnel] server.conf.batch : Activation du service de batch (scripting applicatif)

- Décommenter la ligne "`# server.conf.batch=batch/server.xml`" pour activer le service ;
- Décommenter la ligne "`# server.conf.batch.destPathRoot=${server.work.path}/batchGen`" et spécifier le chemin ou seront créés les fichiers générés via SCENARIbatch

liveCollab

[spécifique DB - optionnel] service liveCollab : module de collaboration entre auteurs

- Décommenter la ligne "`# server.conf.liveCollab=collab/odb/liveCollab.xml`" pour activer le service ;
- Spécifier le port à utiliser : `server.conf.liveCollab.port` =8120 ;
- Paramétrer l'URL d'accès au service à décommenter dans les environnements où liveCollab est lancé via un proxy : `server.conf.liveCollab.connectUrl` =https://[xxx.xxx.xxx.xxx]/scserver41/ws ;
- Décommenter les ligne d'accès à l'outil node JS : `server.conf.liveCollab.nodeJsExePath` =${server.work.path}/nativlib/node.exe et `server.conf.liveCollab.nodeJsLibPath` =${server.work.path}/nodejslib ;
- Paramétrer le path du fichier de log : `server.conf.liveCollab.logPath` =[logPath] /scserver41/scenari-liveCollab.log;

LDAP

[optionnel] LDAP

Pour activer le mode LDAP :

- Déclarer le userMgr :
  - Sur **FS** : `server.conf.usermgr` =usermgr/fs/fsWithLdap.xml ;
  - Sur **DB** : `server.conf.usermgr` =usermgr/odb/odbWithLdap.xml;
- Configurer la connexion LDAP dans les fichiers `fsWithLdap.xml` ou `odbWithLdap.xml`.

Remarques :

- en environnement FS, les utilisateurs ne doivent pas être redéclarés dans scenari ;
- en environnement FS, le rôle est commun à tous les utilisateurs LDAP ;
- en environnement DB, les utilisateurs LDAP doivent être déclarés dans scenari afin de spécifier pour cet utilisateur le ou les rôles utiles.

Ateliers dérivés & brouillon (Spécifique à la version base de données)

[optionnel] server.db.drv : active la possibilité de créer des ateliers dérivés.

Cette option n'existe que dans SCENARIserver (pas dnas SCENARIserverLite).

[optionnel] server.db.drf : active la possibilité de créer des ateliers calque de brouillon.

Cette option n'existe que dans SCENARIserver (pas dnas SCENARIserverLite).

Fichier "advanced/server.xml"

Ce fichier permet de définir des paramètres avancés du serveur

[optionnel] Propriété "sOoLoadWaitFactor"

- Décommenter la ligne `!-- <properties type="Alone" code="com.scenari.s.co.transform.oo.HTransformerOo.sOoLoadWaitFactor" level="2000" >8</properties>--` pour contourner des problèmes de génération de LibreOffice [http://qa.openoffice.org/issues/show_bug.cgi?id=63564] (chargement faussement synchrone du fichier LibreOffice).
- La valeur "8" peut être ajustée en fonction du volume des documents générés et de la puissance du serveur.

[optionnel] Tags "ooConnectionPool"

Ces directives permettent de paramétrer le mode de communication de Scenari avec LibreOffice, et notamment de définir un pool de connexions permettant un usage parallèle de LibreOffice (plusieurs requêtes simultanées) améliorant les performances de générations multiples simultanées.

> Le mode **headless** ne fonctionne pas correctement sous Windows (OO3.3) : le process OO peut se figer dans certains cas.

Exemple de paramétrage sous Linux :

```
1 <declareTag tagName="ooConnectionPool" classParser="com.scenari.s.co.transform.oo.OoConnectionPoolLoader"/>
2 <ooConnectionPool>
3   <ooConnection pipeMode="true" autoLaunch="true" headless="true" profilePath="file:///tmp/LO1"/>
4   <ooConnection pipeMode="true" autoLaunch="true" headless="true" profilePath="file:///tmp/LO2"/>
5 </ooConnectionPool>
```

Exemple de paramétrage sous Windows :

```
1 <declareTag tagName="ooConnectionPool" classParser="com.scenari.s.co.transform.oo.OoConnectionPoolLoader"/>
2 <ooConnectionPool>
3   <ooConnection pipeMode="true" autoLaunch="true" headless="false" profilePath="file:///C:/tmp/OO1"/>
4   <ooConnection pipeMode="true" autoLaunch="true" headless="false" profilePath="file:///C:/tmp/OO2"/>
5 </ooConnectionPool>
```

Windows 64bits

> LibreOffice n'existe qu'en 32bits pour le moment sous Windows. Pour un fonctionnement avec Tomcat 64bits, il est nécessaire de réaliser les actions suivantes :
>
> - déclarer la variable d'environnement UNO_PATH (exemple : `C:\Program Files (x86)\LibreOffice 4.0\program` (ajuster le path vers OO)) ;
> - déclarer une connexion en mode socket dans Scenari via le fichier `[pathSrc]/Vx.y.z/SCENARIserver/conf/advanced/server.xml`.
>
> Exemple :
>
> ```
> 1 <declareTag tagName="ooConnectionPool" classParser="com.scenari.s.co.transform.oo.OoConnectionPoolLoader"/>
> ```

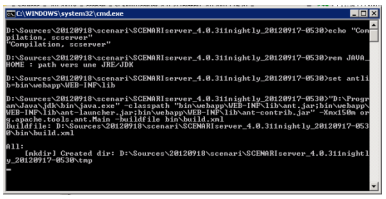
```
<ooConnectionPool>
  <ooConnection host="localhost" port="8244" autoLaunch="true" autoKill="true" headless="false" profilePath="file:///C:/TMP/LO1"/>
</ooConnectionPool>
```

2. Installation

Création de la webApp SCENARIserver

Les sources SCENARIserver doivent avoir été décompréssées au préalable dans le répertoire [pathSrc]/Vx.y.z/SCENARIserver.

- Paramétrer la webApp via les fichier de configuration du répertoire [pathSrc]/Vx.y.z/SCENARIserver/conf (cf section "Configuration") ;
- Exécuter le fichier [pathSrc]/.../makeWar.bat ou [pathSrc]/.../makeWar.sh



Compilation en cours (sous windows)

Le fichier "**scserver40.war**" doit alors avoir été créé dans le répertoire ou [pathSrc]/...

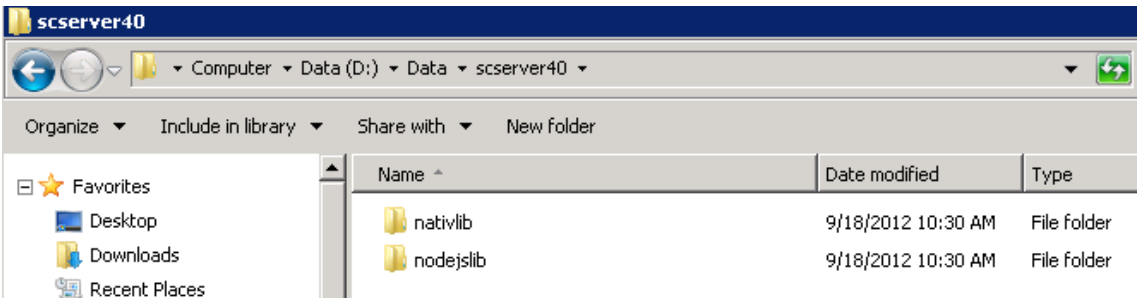
bin 9/18/2012 10:11 AM File folder
conf 9/18/2012 10:11 AM File folder
data 9/18/2012 10:11 AM File folder
license.scenari 9/17/2012 5:34 AM Text Document 4 KB
license.thirdParty 9/17/2012 5:33 AM Text Document 149 KB
license.thirdParty_link 9/17/2012 5:33 AM HTML Document 4 KB
makeWar 9/17/2012 5:30 AM Windows Batch File 1 KB
manual 9/17/2012 5:34 AM Text Document 2 KB
scserver40.war 9/18/2012 10:22 AM WAR File 11,055 KB

Initialisation de l'environnement d'exécution

L'installation et la mise à jour de SCENARIserver impliquent la définition de l'environnement d'exécution.

- Créer le répertoire [pathData] si il n'existe pas ;
- Copier/coller le contenu du répertoire [pathSrc]/Vx.y.z/SCENARIserver/data dans le répertoire [pathData]

Exemple windows



Commentaires généraux

[Spécifique serveur lite]

Les informations d'identifications sont stockées dans le répertoire : [pathData] /users/ac/

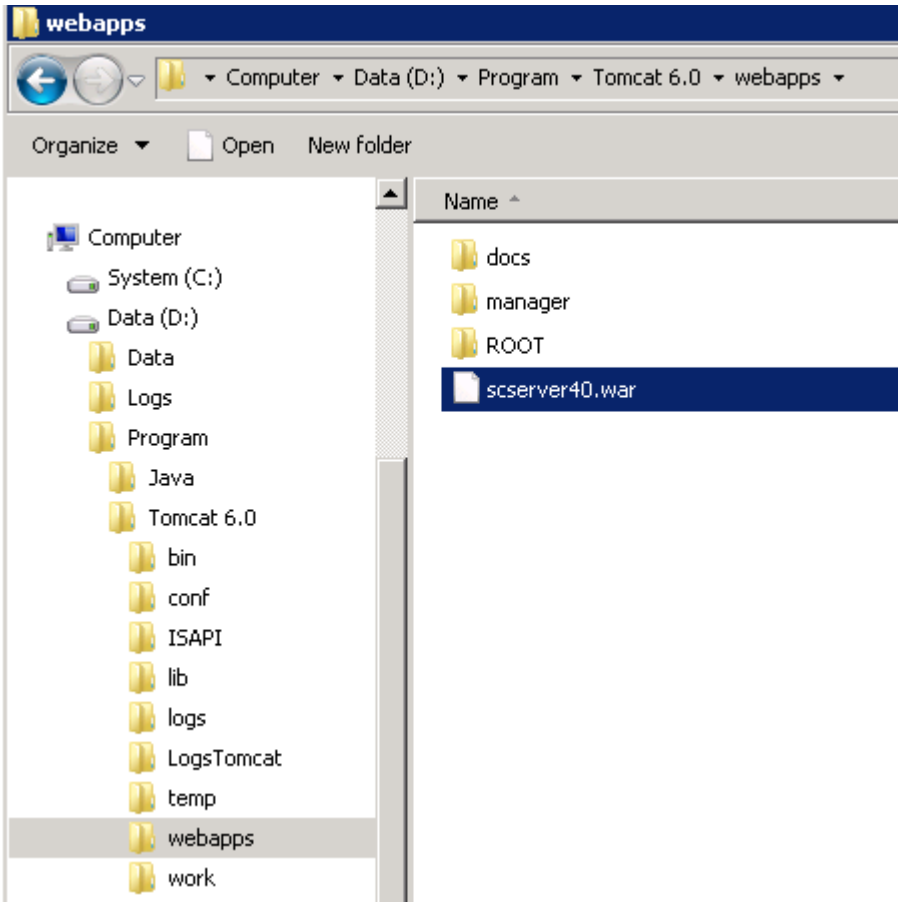
Déploiement de la webApp SCENARIserver

Déploiement du fichier xxx.war précédemment créé dans le serveur d'application.

Convention

Dans cette procédure, xxx désigne le nom de la wepApp produite

- Stopper le serveur d'application si ça n'est pas déjà fait ;
- Copier/coller le fichier xxx.war dans le répertoire [TOMCAT_HOME]/webapps (ou [JBOSS_HOME]/server/scserver/deploy)

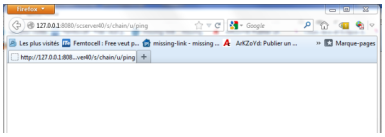


- Supprimer le répertoire `[TOMCAT_HOME]/webapps/xxx` si il existe
- Relancer le serveur d'application

Commentaires généraux

Test de bon fonctionnement

Pour tester la bonne installation de l'application, visitez dans un navigateur web l'URL suivante : http://localhost:8080/scserver41/s/chain/u/ping. Après s'être authentifié, la page web résultante doit être blanche.



Remarque : ajustez les paramètres "8080" (port) et "scserver41"(server.webapp.name) en fonction des paramètres de compilation de la webApp et du paramétrage du serveur d'application.

Login/password par défaut

Le login/password d'administration par défaut est : admin/admin.

Remarque : Celui-ci est modifiable à travers une application cliente connectée au serveur en tant qu'administrateur.

L'application cliente peut maintenant être utilisée, et la connexion vers ce SCENARIserveur (entrepôt) déclarée.

Se référer au manuel de gestion fonctionnelle pour plus de détail sur le paramétrage applicatif auteur.

3. Montée de version

Passage de SCENARIserver4.0 à SCENARIserver4.1

Paramétrage du serveur

Les fichiers de configuration de la webapp n'ont que peu évolué entre SCENARIserver4.0 et SCENARIserver4.1 (hormis une réorganisation mineure).

Néanmoins, de façon générale pour toute nouvelle version, il convient de **reporter dans ces nouveaux fichiers** de configuration **les ajustements** que vous avez réalisé pour votre précédent système.

Évolution de la configuration LDAP [DB uniquemennt]

L'évolution du paramétrage de la connexion LDAP nécessite d'opérer un ajustement sur les différents utilisateurs déjà déclarés.

Pour réaliser cela automatiquement, si un contexte LDAP était déjà actif dans votre système en 4.0, il vous faudra exécuter le script batch " `apps/dbMigrFrom40To41` " présent dans l'application SCENARIbatch *(cf. SCENARIbatch)* [p.43].

Données utilisateur

SCENARIserver [db]

La **compatibilité ascendante** est assurée. Ainsi, dans un environnement 4.1, il suffit de référencer le **même jeu de données** (variable de configuration *`server.work.path`*) que celui précédemment exploité en 4.0 pour qu'une **transformation automatique** de celui-ci soit opérée lors du premier lancement.

Répertoires concernés :

- `[server.work.path]/javaserver/db` ;
- `[server.work.path]/javaserver/blobs` ;
- `[server.work.path]/javaserver/privData` ;

Fichier txLog [db]

La structure du fichier de log `[server.work.path]/javaserver/db/txlog.otx` a changé dans SCENARI4.1. Ce fichier, exploité uniquement pour enregistrer des logs transactionnels, peut être supprimé avant la migration SCENARIserver4.1.

Si cette suppression n'est pas réalisée, l'erreur ci-dessous peut se produire, et sera à ignorer (aucune incidence fonctionnelle) :

```
INFO: Server startup in 4225 ms
15 juil. 2014 12:14:56 com.orientechnologies.common.log.OLogManager log
ATTENTION: Error on opening the txlog file 'File: txlog.otx os-size=1000000,
stored=998976, filled=998976, max=536870912', reset it
```

```
4 com.orientechnologies.common.io.OIOException: You cannot access outside the
  file size (998976 bytes). You have requested portion -83439842--83439841 bytes.
  File: File: txlog.otx os-size=1000000, stored=998976, filled=998976, max=536870912
5  at com.orientechnologies.orient.core.storage.fs.OAbstractFile.checkRegions
  (OAbstractFile.java:363)
6  at com.orientechnologies.orient.core.storage.fs.OFileClassic.checkRegions
  (OFileClassic.java:230)
7  at com.orientechnologies.orient.core.storage.fs.OFileClassic.readByte
  (OFileClassic.java:120)
8  at com.orientechnologies.orient.core.storage.impl.local.OTxSegment.
  scanForTransactionsToRecover(OTxSegment.java:233)
9  at com.orientechnologies.orient.core.storage.impl.local.OTxSegment.
  recoverTransactions(OTxSegment.java:200)
10 at com.orientechnologies.orient.core.storage.impl.local.OTxSegment.open
  (OTxSegment.java:87)
```

- Avant de réaliser cette montée de version, il convient de réaliser une sauvegarde complète des données ;
- La phase de transformation automatique des données peut durer **plusieurs minutes** en fonction de la taille de la base de données. Son début est signifié par les lignes suivantes dans le fichier de log :

```
1 1--- Info : Tue Jul 15 11:54:30 CEST 2014 (DbScheduler:repos) ---
2 Start database migration DbVersion 'core' from 0 to 1. (class eu.scenari.
  orient.init.migr.DbMigrCore0_1)
3
4
5 1--- Info : Tue Jul 15 11:54:35 CEST 2014 (DbScheduler:repos) ---
6 Start rebuild Db : repos - local-sc:D:/scenari/data/scserver41/javaserver/db
```

  La fin (et la réussite) de cette étape seront indiqué par ces lignes :

```
1 1--- Info : Tue Jul 15 11:56:13 CEST 2014 (DbScheduler:repos) ---
2 End rebuild Db: repos in 98280ms
3
4
5 1--- Info : Tue Jul 15 11:56:13 CEST 2014 (DbScheduler:repos) ---
6 End database migration DbVersion 'core' from 0 to 1
```

  Un arrêt du serveur pendant cette phase de migration peut conduire à une perte des données, et nécessitera l'utilisation d'un backup.
- Les données de la 4.1 ne peuvent pas être utilisées dans un contexte 4.0 (pas de compatibilité descendante) ;

SCENARIserverLite

Aucune transformation de donnée n'est opérée : les structures de donnée sont **iso-fonctionnelles**.

Répertoires concernés :

- `[server.work.path]/javaserver/wsps` ;
- `[server.work.path]/javaserver/privData` ;

Données applicatives

Répertoire temporaire

Le répertoire temporaire paramétré via la variable *`server.temp.path`* peut être purgé.

Modèle documentaire

Le ou les modèles documentaires (wsppack) installés et exploités dans une version X.Y de scenari sont **incompatibles** avec toute version X.Y+1 (le système l'indiquera clairement).

Il convient donc d'**installer** sur SCENARIserver4.1 **le ou les modèles documentaires dédiés**.

Répertoires concernés à purger donc :

- `[server.work.path]/javaserver/pack` ;
- `[server.work.path]/javaserver/updt` ;

Chapitre 6

Installation/Configuration de SCENARIserver sous Debian

scenari-platform.org fournit SCENARIserver sous format ***DEB*** pour Debian et Ubuntu. Cette méthode d'installation de SCENARIserver apporte des avantages mais aussi des inconvénients.

Avantages

- Une installation simplifiée utilisant toute la force du système de paquetage Debian pour garantir que tous les prérequis seront également installés, en particulier OpenJDK et Tomcat / Jetty mais également Imagemagick et ffmpeg.
- Un SCENARIserver utilisant le paramétrage par défaut est opérationnel en quelques minutes.
- SCENARIserver sera mis à jour automatiquement dès qu'une nouvelle version est disponible sur des dépôts.

Inconvénients & Points à contrôler

- Cette méthode ne permet pas l'installation de plusieurs webapps SCENARIserver dans le même serveur d'application. Votre serveur devient donc un SCENARIserver unique.

  Si vous avez besoin de ce type d'installation alors vous devez vous reporter à la procédure l'installation générique.
- Ce mode d'installation ne vous dispense pas d'effectuer les étapes de Configuration système *(cf. Configuration système)* [p.10] et de Configuration de Tomcat *(cf. Configuration de Tomcat)* [p.13].

1. Installation

Installer SCENARIserver sur Debian / Ubuntu

Pré-requis

Vous devez disposer des droits d'administration.

Choix du serveur d'application

SCENARIserver peut être intégré dans l'un des trois serveur d'applications suivants :

- Tomcat 6 ;
- Tomcat 7 ;
- Jetty.

Il existe un paquet spécifique à chacun de ces serveurs.

Le serveur d'application préconisé pour SCENARIserver est actuellement **Tomcat 6**, Peu d'installations ont été faites sous d'autres serveurs d'applications.

Pour installer l'application SCENARIserver, procéder comme suit :

- Ouvrir un terminal.
- Ajouter le dépôt concernant votre distribution.

Ubuntu

```
user@computer:~$ echo "deb http://scenari-platform.org/deb `lsb_release -sc` main" | sudo tee -a /etc/apt/sources.list.d/scenari.list
```

Ou vous pouvez également ajouter la ligne deb approprié via Logiciels & mises à jour dans les paramètres système :

14.10 (`Utopic`) : `deb http://scenari-platform.org/deb utopic main`

14.04 LTS (`Trusty`) : `deb http://scenari-platform.org/deb trusty main`

12.04 LTS (`Precise`) : `deb http://scenari-platform.org/deb precise main`

Debian

Versions supportées : `sid`, `jessie` ou `wheezy`.

Ajoutez la ligne deb approprié à vos sources d'applications :

`sid` : `deb http://scenari-platform.org/deb sid main`

`jessie` : `deb http://scenari-platform.org/deb jessie main`

`wheezy` : `deb http://scenari-platform.org/deb wheezy main`

Veuillez consulter la documentation du gestionnaire de paquets de votre distribution pour plus d'information (Ubuntu [https://help.ubuntu.com/lts/ubuntu-help/addremove-sources.html]).

- Ajouter la clé d'authentification.

```
user@computer:~$ wget -O- http://scenari-platform.org/deb/scenari.asc | sudo apt-key add -
```

Cette clé permet au système de vérifier l'authenticité et l'intégrité des paquets proposés.

Vous pouvez installer cette clé via votre gestionnaire de paquet graphique en la téléchargeant au préalable.

Il est cependant possible d'utiliser un dépôt sans ajouter la clé d'authentification. Mais ce n'est pas recommandé.

Plus d'information à ce sujet et installation manuelle dans la documentation ubuntu [http://doc.ubuntu-fr.org/tutoriel/comment_modifier_sources_maj#gerer_les_cles_d_authentification].

- Mettre à jour la liste des paquets disponible

```
user@computer:~$ sudo apt-get update
```

- Installer SCENARIserver à l'aide du paquet instrumentalisant l'intégration souhaitée.

  Vous devez maintenant choisir votre serveur d'application (Tomcat 6, Tomcat 7 ou Jetty)

  Pour SCENARIserver 4.1 sous Tomcat 6 :

```
user@computer:~$ sudo apt-get install scenariserver4.1-tomcat6
```

  Pour SCENARIserverLite 4.1 sous Tomcat 6 :

```
user@computer:~$ sudo apt-get install scenariserverlite4.1-tomcat6
```

  SCENARIserver [DB]

```
user@computer:~$ apt-cache search SCENARIserver4.1
scenariserver4.1 - SCENARIserver 4.1 - Core web-app files.
scenariserver4.1-jetty - SCENARIserver 4.1 - Jetty integration.
scenariserver4.1-tomcat6 - SCENARIserver 4.1 - Tomcat6 integration.
scenariserver4.1-tomcat7 - SCENARIserver 4.1 - Tomcat7 integration.
```

  SCENARIserverLite [Lite]

```
user@computer:~$ apt-cache search SCENARIserverLite4.1
scenariserverlite4.1 - SCENARIserverLite 4.1 - Core web-app files.
scenariserverlite4.1-jetty - SCENARIserverLite 4.1 - Jetty integration.
scenariserverlite4.1-tomcat6 - SCENARIserverLite 4.1 - Tomcat6 integration.
scenariserverlite4.1-tomcat7 - SCENARIserverLite 4.1 - Tomcat7 integration.
```

- Relancer le serveur d'application.

  Pour Tomcat 6 :

```
user@computer:~$ sudo service tomcat6 restart
```

Résultat

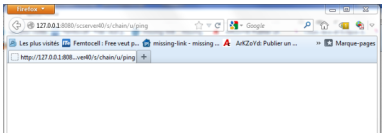
L'installation de SCENARIserver est finie. Celui-ci est maintenant opérationnel avec le paramétrage par défaut.

Commentaires généraux

Test de bon fonctionnement

Pour tester la bonne installation de l'application, visiter dans un navigateur web l'URL suivante : http://[xxx.xxx.xxx.xxx]:8080/scenariserver4.1/s/chain/u/ping.

Après s'être authentifié, la page web résultante doit être blanche.



Remarque : ajustez les paramètres "8080" (port) et "scenariserver4.1"(server.webapp.name) en fonction des paramètres de compilation de la webApp et du paramétrage du serveur d'application.

Login/password par défaut

> Le login/password d'administration par défaut est : admin/admin.
>
> Remarque : Celui-ci est modifiable à travers une application cliente connectée au serveur en tant qu'administrateur.

> L'application cliente peut maintenant être utilisée, et la connexion vers ce SCENARIserver (entrepôt) déclarée.
>
> Se référer au manuel de gestion fonctionnelle pour plus de détail sur le paramétrage applicatif auteur.

2. Configuration

Pour commencer...

La webApp SCENARIserver est fortement paramétrable en fonction du contexte d'utilisation et des fonctionnalités désirées.

Le paramétrage d'un SCENARIserver installé sous Debian / Ubuntu en DEB se trouve dans `/etc/scenariserver4.1/`.
Ce dossier contient :

- un fichier `init.xml` contenant la configuration actuellement active ;
- un dossier `conf` contenant les ressources nécessaires à la construction du fichier `init.xml`.

> **Ne pas éditer le fichier `init.xml` directement mais modifier les fichiers dans le dossier `conf`.**
>
> Une fois les modifications apportées, la commande `scenariserver4.1-cfg` permet de reconstruire le fichier `init.xml`.

Dans les parties suivantes, nous allons décrire une partie du paramétrage disponible dans le dossier `/etc/scenariserver4.1/conf`.

Fichier "main.property "

Chemins d'accès

[requis] server.work.path : Répertoire principal des données

Cette variable définit le répertoire `[pathData]` utilisé dans le présent document à diverses reprises.

Exemple : **server.work.path**=E:/data/scserver41

[optionnel] server.temp.path : Répertoire de travail (backup inutile)

Cette variable définit le répertoire [pathData] utilisé dans le présent document à diverses reprises.
Exemple : **server.temp.path**=E:/tmp/scserver41

[optionnel] server.gen.path : Répertoire racine de sortie des générateurs

Cette variable définit le répertoire [pathData] utilisé dans le présent document à diverses reprises.
Exemple : **server.gen.path**=E:/data/scserver41/javaserver/gen

[optionnel] server.log.path : Répertoire des logs scenari

Cette variable définit le fichier de log scenari.
Exemple : **server.log.path**=[pathLog]/[webAppName]/scenari_[date].log
Remarque : penser à retirer le caractère # de mise en commentaire de la variable server.log.path

Sécurité

[optionnel] server.conf.enhancedSecurity : Activation de la sécurité avancée

Cette variable permet d'activer un niveau de sécurité accru
Remarque : consommateur de ressources

Configuration du mode batch

[optionnel] server.conf.batch : Activation du service de batch (scripting applicatif)

- Décommenter la ligne "`# server.conf.batch=batch/server.xml`" pour activer le service ;
- Décommenter la ligne "`# server.conf.batch.destPathRoot=${server.work.path}/batchGen`" et spécifier le chemin ou seront créés les fichiers générés via SCENARIbatch

liveCollab

[spécifique DB - optionnel] service liveCollab : module de collaboration entre auteurs

- Décommenter la ligne "`# server.conf.liveCollab=collab/odb/liveCollab.xml`" pour activer le service ;
- Spécifier le port à utiliser : `server.conf.liveCollab.port` =8120 ;
- Paramétrer l'URL d'accès au service à décommenter dans les environnements où liveCollab est lancé via un proxy : `server.conf.liveCollab.connectUrl` =https://[xxx.xxx.xxx.xxx]/scserver41/ws ;
- Décommenter les ligne d'accès à l'outil node JS : `server.conf.liveCollab.nodeJsExePath` =${server.work.path}/nativlib/node.exe et `server.conf.liveCollab.nodeJsLibPath` =${server.work.path}/nodejslib ;

- Paramétrer le path du fichier de log : `server.conf.liveCollab.logPath` =[logPath] /scserver41/scenari-liveCollab.log;

LDAP

[optionnel] LDAP

Pour activer le mode LDAP :

- Déclarer le userMgr :
  - Sur **FS** : `server.conf.usermgr` =usermgr/fs/fsWithLdap.xml ;
  - Sur **DB** : `server.conf.usermgr` =usermgr/odb/odbWithLdap.xml;
- Configurer la connexion LDAP dans les fichiers `fsWithLdap.xml` ou `odbWithLdap.xml`.

Remarques :

- en environnement FS, les utilisateurs ne doivent pas être redéclarés dans scenari ;
- en environnement FS, le rôle est commun à tous les utilisateurs LDAP ;
- en environnement DB, les utilisateurs LDAP doivent être déclarés dans scenari afin de spécifier pour cet utilisateur le ou les rôles utiles.

Ateliers dérivés & brouillons

[optionnel] server.db.drv : active la possibilité de créer des ateliers dérivés.

Cette option n'existe que dans SCENARIserver (pas dnas SCENARIserverLite).

[optionnel] server.db.drf : active la possibilité de créer des ateliers calque de brouillon.

Cette option n'existe que dans SCENARIserver (pas dnas SCENARIserverLite).

Fichier "advanced/server.xml"

Ce fichier permet de définir des paramètres avancés du serveur

[optionnel] Propriété "sOoLoadWaitFactor"

- Décommenter la ligne `!-- <properties type="Alone" code="com.scenari.s.co.transform.oo.HTransformerOo.sOoLoadWaitFactor" level="2000">8</properties>--` pour contourner des problèmes de génération de LibreOffice [http://qa.openoffice.org/issues/show_bug.cgi?id=63564] (chargement faussement synchrone du fichier LibreOffice).
- La valeur "8" peut être ajustée en fonction du volume des documents générés et de la puissance du serveur.

[optionnel] Tags "ooConnectionPool"

Ces directives permettent de paramétrer le mode de communication de Scenari avec LibreOffice, et notamment de définir un pool de connexions permettant un usage parallèle de LibreOffice (plusieurs

requêtes simultanées) améliorant les performances de générations multiples simultanées.

> Le mode **headless** ne fonctionne pas correctement sous Windows (OO3.3) : le process OO peut se figer dans certains cas.

Exemple de paramétrage sous Linux :

```
<declareTag tagName="ooConnectionPool" classParser="com.scenari.s.co.transform.oo.OoConnectionPoolLoader"/>
<ooConnectionPool>
  <ooConnection pipeMode="true" autoLaunch="true" headless="true" profilePath="file:///tmp/LO1"/>
  <ooConnection pipeMode="true" autoLaunch="true" headless="true" profilePath="file:///tmp/LO2"/>
</ooConnectionPool>
```

La commande `scenariserver4.1-cfg`

Les version DEBs de SCENARIserver sont fournies avec un script de maintenance : `scenariserver4.1-cfg`.
Celui-ci permet de reconfigurer ou de réinitialiser la configuration et d'effectuer une sauvegarde simple de SCENARIserver [DB]

Appliquer des modifications de configuration (`reconfig`)

Une fois la configuration de SCENARIserver modifiée il faut re-générer le fichier `/etc/scenariserver4.1/init.xml` puis relancer le serveur d'application pour que les modifications soient prises en compte.

- Générer le fichier de configuration `/etc/scenariserver4.1/init.xml`.

  ```
  user@computer:~$ sudo scenariserver4.1-cfg reconfig
  ```

  Cette commande sauvegarde le fichier `/etc/scenariserver4.1/init.xml` actuel en `/etc/scenariserver4.1/init.xml.bkp` puis l'écrase avec une nouvelle version construite à partir des éléments dans `/etc/scenariserver4.1/conf`.

- Relancer le serveur d'application.

  Pour Tomcat 6 :

  ```
  user@computer:~$ sudo service tomcat6 restart
  ```

Réinitialiser la configuration (`reset`)

À tout moment, il est possible de supprimer toute modification de la configuration de SCENARIserver et de revenir au paramétrage par défaut.

Cette procédure ne peut être annulée et effacera toute modification apportée à la configuration de SCENARIserver.

- Réinitialiser la configuration dans `/etc/scenariserver4.1/`.

```
user@computer:~$ sudo scenariserver4.1-cfg reset
```

Cette commande écrase le dossier `/etc/scenariserver4.1/conf` avec les fichiers originaux. `scenariserver4.1-cfg` propose ensuite de lancer la reconstruction du fichier `/etc/scenariserver4.1/init.xml`

- Relancer le serveur d'application.

Pour Tomcat 6 :

```
user@computer:~$ sudo service tomcat6 restart
```

Sauvegarder les données (`backup`) [DB]

`scenariserver4.1-cfg` propose une fonction simple de sauvegarde de la base de données de SCENARIserver.

Cette commande dépend de la bonne configuration de `scenariserver4.1-cfg` qui s'effectue dans le fichier : `/etc/scenariserver4.1/cfg.conf` :

```
# scenariserver4.1-cfg script config file
webappUrl=http://localhost:8080/scenariserver4.1
webappUser=admin
webappPasswd=admin
dataFolder=/var/lib/scenariserver4.1/javaserver
backupFolder=/var/lib/scenariserver4.1/backup
```

- `webappUrl` : url dans la webapp SCENARIserver telle que vue depuis le serveur
- `webappUser`, `webappPasswd` : identifiant et mot de passe permettant de se connecter à SCENARIserver
- `dataFolder` : dossier source contenant la base de données
- `backupFolder` : dossier de destination de la sauvegarde

IMPORTANT : ne pas oublier d'adapter ces valeurs en fonction de votre paramétrage de SCENARIserver.

Cette sauvegarde s'effectue forcément à chaud.

- S'assurer que le serveur d'application est bien lancé et que SCENARIserver est bien accessible.
- Lancer la commande de sauvegarde :

```
user@computer:~$ sudo scenariserver4.1-cfg backup
```

Résultat

Une copie de la base de données est maintenant disponible dans le dossier `backupFolder` (par défaut `/var/lib/scenariserver4.1/backup`).

> Vous pouvez incorporer cette procédure dans votre stratégie de sauvegarde de vos serveurs. Le contenu de `backupFolder` peut alors être récupéré par un système tiers tel que ***backuppc*** ou ***backula*** etc.

Chapitre 7

Monitoring

SCENARI4.1 offre des possibilités de surveillance par des environnements de monitoring (nagios, ...), afin de suivre la bonne santé du système.

Nous recommandons vivement la mise en place de ces sondes de contrôle dans les système mis en production.

Nous préconisons au minimum la déclaration des deux services suivants :

Contrôle de la webApp SCENARIserver

Service check_http pour contrôler que la webapp SCENARIserver fonctionne

url : https://[urlWebapp]/s/chain/u/ping

user : [scuserSystem]

password : [mot de passe [scuserSystem]]

Assertion sur la réponse HTTP :

- 200 : succès ;
- 503 : warning ;
- erreur dans tous les autres cas ;

Exemple de paramétrage Nagios :

Déclaration d'une nouvelle commande :

define command{

command_name ping_scserver_ssl

command_line /usr/lib/nagios/plugins/check_http -S -H '$ARG1$' -u '/$ARG2$/s/chain/u/ping' -a $ARG3$:$ARG4$

}

Déclaration d'un nouveau service :

define service{

use generic-service

host_name [serverHostname]

service_description SCserver-coperia2

servicegroups scservers

check_command ping_scserver_ssl![serverHostname]![nameWebapp]![scuserSystem]![mot de passe [scuserSystem]]!

}

Contrôle d'intégrité des données du SCENARIserver [DB]

Service check_http pour contrôler la cohérence des données

url : https://[urlWebapp]/s/chain/u/adminOdb?cdaction=CheckAuto

user : [scuserSystem]

password : [mot de passe [scuserSystem]]

Assertion sur la réponse HTTP :

- 200 : succès ;
- 503 : warning ;
- erreur dans tous les autres cas ;

un check par 24h (lors d'une activité applicative faible)

Exemple de paramétrage Nagios :

Déclaration d'une nouvelle commande :

define command{

command_name ping_scserverdata_ssl

command_line /usr/lib/nagios/plugins/check_http -S -H '$ARG1$' -u '/$ARG2$/s/chain/u/adminOdb?cdaction=CheckAuto' -a $ARG3$:$ARG4$

}

Déclaration d'un nouveau service :

define service{

use generic-service

host_name [serverHostname]

service_description SCserver-coperia2

servicegroups scservers

check_interval 1440

retry_interval 30

check_command ping_scserverdata_ssl![serverHostname]![nameWebapp]![scuserSystem]![mot de passe [scuserSystem]]!

}

Autres contrôles utiles

- Exécution des backups ;
- Place disque ;
- Ram utilisée ;
- ...

Chapitre 8

SCENARIbatch

SCENARIbatch propose des librairies utilitaires permettant de piloter - en dehors de l'interface cliente - SCENARIserver : gestion des items, déclaration massive d'utilisateurs, déclenchement de génération, ...

Les librairies proposées sont codées dans le langage de scripting ANT [http://ant.apache.org/] .

1. Installation

Installation

Convention

[pathScBatch]	Chemin d'installation de SCENARIbatch

Activation du service batch

> Afin de rendre utilisables les directives SCbatch, il est impératif d'activer le service batch lors du paramétrage et de la compilation de la webapp.

- Décompresser le fichier **SCENARIbatch.zip** téléchargé dans le répertoire [pathScBatch].

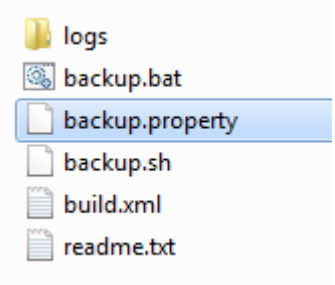
2. Sauvegarde / Restauration [DB]

> La mise en place d'un système de sauvegarde des données est fortement recommandée sur SCENARIserverLite (fileSystem), mais devient **indispensable** dans l'environnement base de données de SCENARIserver4.
>
> Effectivement, ce type d'architecture logicielle est sensible aux instabilités système qui peuvent compromettre l'intégrité des données (arrêt non programmé de la machine, ...).

Sauvegarde à chaud (système en fonctionnement)

Sauvegarde : configuration

- Ouvrir le fichier [pathScBatch]/apps/backup/backup.property ;



- Spécifier les propriétés du programme dans ce fichier, et l'enregistrer ;

[requis] program.webapp.url : URL d'accès à la webApp

Exemple :

program.webapp.url=http://127.0.0.1:8080/scserver41

[requis] program.user : login d'administration

Exemple :

program.user=admin

[requis] program.password : password d'administration

Exemple :

program.password=admin

[requis] program.logs.dir : Chemin absolu vers le répertoire de log

Exemple :

program.logs.dir=[pathLog]/scBatch

Processus de sauvegarde

SCENARIserverDb intègre un système de base de données orienté graphe de documents. Ainsi, il est important de lancer les processus de backup sur un état stable de la base, afin d'en maintenir l'intégrité référentielle lors des restaurations.

Pour ce faire, deux commandes sont à exécuter :

- **avant la sauvegarde des fichiers** : permet de passer la base de donnée dans un statut "backup en cours" mémorisant ainsi les suppressions de fichier (blobs), et mettant à disposition des fichiers d'index complets et corrects ;
- **après la sauvegarde** : permet de remettre le système dans un état de fonctionnement normal (suppression du statut "backup en cours""

> La précédente procédure s'exécute avec le SCENARIserver lancé, et ne gène en rien la production.

Exécuter le script [pathScBatch]/apps/backup/backup.bat pre , ou [pathScBatch]/apps/backup/backup.sh pre ;

Cette exécution doit se faire dans le répertoire `[pathScBatch]/apps/backup`.

- Sauvegarder les fichiers de donnée via l'outil de backup au choix

Répertoires à sauvegarder

- Base de données :
  - [pathData]/javaserver/blobs ;
  - [pathData]/javaserver/backup ;
- [optionnel] Environnement de travail : [pathData]/javaserver/privData ;
- [optionnel] Modèles documentaires installés : [pathData]/javaserver/updt et [pathData]/javaserver/pack ; ;
- [optionnel] Générations précédentes : [pathData]/javaserver/gen ;

Les différents types de backup de fichiers usuels sont utilisables : totale, incrémentale, différentielle.

- Exécuter le script [pathScBatch]/apps/backup/backup.bat post , ou [pathScBatch]/apps/backup/backup.sh post ;

OU Sauvegarde système arrété

Seule l'étape "Sauvegarder les fichiers de donnée via l'outil de backup au choix" de la procédure ci-dessus est à mettre en oeuvre.

Restauration

Processus de restauration

Un incident système grave compromettant l'intégrité référentielle de la base de donnée peut conduire à la nécessité de restauration d'une sauvegarde.

Dans cette procédure, on considérera le répertoire [backupDir] comme répertoire contenant une sauvegarde complète des données utilisateur.

- Arrêter le serveur d'application
- Restaurer les répertoires sauvegardés

Restauration de la base de donnée

- Supprimer le répertoire [pathData]/javaserver/db ;
- Copier le contenu du répertoire [backupDir]/javaserver/backup dans le répertoire [pathData]/javaserver/db ;

> **Restauration des autres répertoires**
>
> Copier/coller le contenu des autres répertoires sauvegardés (privData, updt, pack, gen, blobs, ...) de [backupDir] dans les répertoire du même nom de [pathData].

- Relancer le serveur d'application

Chapitre 9

Proxy [DB]

Le cœur de SCENARIserver est une webapp qui réside dans un serveur de servlet tel que Tomcat : la plupart de la communication entre le serveur et les clients se fait en http ou https.

SCENARIserver comporte un module qui prend en charge tous les aspects collaboratifs entre auteurs. Ce module est un serveur nodejs qui communique par défaut avec le client en websocket sur un port dédié.

Nginx [http://nginx.org/en/docs/http/websocket.html] et Apache [http://httpd.apache.org/docs/2.4/mod/mod_proxy_wstunnel.html] sont des serveurs Web HTTP / HTTPS qui permettent de faire passer les deux flux (http et websocket) par un port unique.